Dynamic discovery of e-services: a Description Logics based approach

C. Rey

LIMOS, Université Blaise Pascal, France (rey@isima.fr)

Motivation

The internet are revolutionizing the way companies interact with their supplier, partners and clients. In the last decade, the number and type of on-line services increased considerably and leads to a new form of automation, namely B2B and B2C e-commerce. A recent industrial initiative envisions a new paradigm for electronic commerce in which applications are wrapped and presented as integrated electronic services (E-services) [1, 2]. An e-service can be defined as an application made available via the Internet by a service provider, and accessible by clients [8]. Examples of e-services currently available range from on-line travel reservation or banking services to entire business functions of an organization. What makes such a vision attractive is that e-services are capable of intelligent interaction by being able to discover and negotiate with each other, compose themselves into more complex services, etc...([7, 1]). The aim is to always try to adapt to the clients' needs, changing from the "do it yourself" to the "do it for me" philosophy.

In this context, this work focuses on the problem of the dynamic discovery of e-services : given a user query and a bunch of e-services, how to discover combinations of e-services that can be used to answer parts or the whole of the query ? As it has been initiated in [11], we propose to use Description Logics and their reasoning capabilities to achieve this problem. Description Logics perfectly cope with the spirit of the Semantic Web by defining semantically rich languages to describe e-services and by providing logical reasoning techniques to manipulate them. The main contribution of this work is the definition of a new complex Description Logics reasoning, based on more basic ones, namely subsumption and difference, that allows the dynamic discovery of e-services.

As an application, this complex reasoning for dynamic selection of e-services will be implemented in the MKBEEM projet[1]. It will be used in association with the PICSEL system ([10]) which implements another complex Description Logics reasoning, namely the rewriting of queries using views ([5]). The complementary roles of these two complex logical reasoning will constitute the Description Logics core for query processing in the MKBEEM e-commerce platform.

[1]MKBEEM stands for Multilingual Knowledge Based European Electronic Marketplace (IST-1999-10589, 1st Feb. 2000 - 1st Aug. 2002).

Statement of the problem

Assume that we have a terminology $\mathcal{T}$ of e-services S_i expressed in a description logic $\mathcal{L}$, and a user query Q expressed in the same language. Intuitively, the problem is to find in $\mathcal{T}$ a set E of e-services that best answers Q. This implies 2 issues to be addressed:

(1) The first issue is to be able to compute and measure the difference between the query and any set of e-services.

Formally, it amounts to use a difference operator associated with the notion of size of a description. Whereas the size of a description is a well-known notion (see [3]), the difference of two descriptions is not often used. In [14], Teege defines a purely semantic difference operator as follows:
$B - A = Max_{\sqsupseteq}\{C|C \sqcap A \equiv B\}$ with $B \sqsubseteq A$
The main advantage of this definition is that the semantics of the operator is very clear: $B - A$ is the set of the most general concepts that give B when intersected with A. Its main drawback is that, for quite expressive languages, the difference may be non semantically unique. However, languages for which this difference is unique and can be syntactically computed (as a sort of set-theoretical difference) have been characterized in [14]: an example language that is given by Teege is called $\mathcal{L}_1$ and allows atoms for concepts, roles, and features, together with the following constructors:

- $\sqcap$, $\sqcup$, $\top$, $\perp$, $(\exists R : C)$, $(\exists f : C)$,$(\geq nR)$ for concepts,
- $\perp$, $\circ$, $|$ for roles,
- $\perp$, $\circ$ for features.

$\mathcal{L}_1$ appears to be expressive enough to be useful in practice.

(2) The second issue is, knowing the difference operator, to be able to find a set of e-services that best covers the query, which means a set of e-services that shares the most information possible with the query and brings the least extra information compared to the query.

This amounts to the definition of the notion of *best cover* of a concept-description using a terminology: in fact, as the rewriting queries using views problem, this problem of finding best covers can be formulated as a new instance of the theoretical framework of the rewriting of concept using a terminology problem (see [3]). The following definition formalizes the notion of best cover for languages $\mathcal{L}$ for which Teege's difference is semantically unique and can be syntactically computed. Other languages will be discussed later.

Definition A best cover E of a $\mathcal{L}$-concept description Q using a terminology $\mathcal{T}$ is a conjunction of some names S_i of defined concepts taken from $\mathcal{T}$ such that:

a) $Q - lcs(Q, E) \not\equiv Q$

b) $Q \sqcap E \not\equiv \bot$

c) $(|Q - lcs(Q, E)|, |E - lcs(Q, E)|)$ is minimal with relation to the lexicographical order.

where lcs stands for least common subsumer: $lcs(C, D)$ is the least general concept description that is more general than both C and D.

We call the problem of computing all the best covers the **best covering problem**.

Informally:

a) This condition ensures that the query Q shares some common information with the cover E. Please note that for some languages this condition is not equivalent to $lcs(Q, E) \not\equiv \top$.

b) This condition ensures that Q and E are consistent with each other.

c) This condition ensures that the information of Q not covered by E is the smallest possible, and that the information brought by E that is not in Q is also the smallest possible.

Achieved work

Up to now, the main results that have been reached are the precise formalization of the problem for the previous languages and the proposition of an algorithm to solve it:

- We have shown that the problem of computing the best covers of a query using a terminology amounts to the problem of computing the minimal transversals with a minimal cost of a hypergraph. That allows us to reuse all algorithmic material that exists in this domain.

- The best covering problem has been shown to be **NP**-Hard by polynomially reducing the problem of finding a minimal cardinality transversal of a hypergraph into it.

- The classical algorithm to compute minimal transversals (see [6] or [9]) has been adapted by an added Branch and Bound step and some optimizations.

- An implementation in Java is about to be achieved. This includes:
  - the reading of an XML file containing a terminology of concepts definitions (for the e-services S_i) and a concept description Q (for the query),
  - the normalization of these descriptions,
  - the building of the corresponding hypergraph

– and the algorithm to compute minimal transversals with a minimal cost.

Future efficiency measures are planned to evaluate the practical behaviour of this algorithm.

Work in progress

A positioning study of the problem in the industrial context of e-commerce ([8], [7], [4]) is in progress. In this context, two issues appear important: where does our dynamic discovery of e-services is located in the whole e-services processing, and which expressivity levels are commonly used or required to describe e-services.

Concerning the first problem, our work does correspond to what is called the "dynamic discovery of e-services" in [7]. So, from a B2C point of view, what we propose is the first step before the dynamic composition of e-services. Moreover, from a B2B point of view, it could be used as a way to help services providers to design complex e-services from basic ones.

The second problem is motivated by the following question: how can we extend our approach to languages that are already used on the web ? By regarding specifications like UDDI [2], or DAML+OIL which is a new description logic for the semantic web ([12]), one can say that languages used to describe e-services generally have a high level of expressivity, but consequently have increased complexity reasonings. For example Teege's difference may be non unique.

In his context, we are extended our approach to the quite expressive language $\mathcal{ALN}$. Indeed, it seems to us that $\mathcal{ALN}$ is a good trade-off between a more expressive language, useful in real applications, and a language for which subsumption is not too complicated to handle. The main problem with $\mathcal{ALN}$ is that Teege's difference is semantically non unique. But using the structural characterization of subsumption (see [13]) allows to define a restricted difference operator, by avoiding meaningless decompositions of $\perp$ into conjunctions of non inconsistent concepts, which is semantically unique. This restricted operator makes the definition of best covers valid for $\mathcal{ALN}$. Once this restricted operator will be precisely defined, we will be able to look for an algorithm (possibly an extension to the hypergraph approach) to compute the best covers for $\mathcal{ALN}$-queries and $\mathcal{ALN}$-terminologies of e-services.

References

[1] Data Engineering Bulletin: Special Issue on Infrastructure for Advanced E-Services. 24(1), IEEE Computer Society, 2001.

[2] The VLDB Journal: Special Issue on E-Services. 10(1), Springer-Verlag Berlin Heidelberg, 2001.

[2] http://www.uddi.org/

[3] F. Baader, R. Küsters, and R. Molitor. Rewriting concepts using terminologies – revisited. Report 00-04, LTCS, RWTH Aachen, Germany, 2000. See http://www-lti.informatik.rwth-aachen.de/Forschung/Reports.html.

[4] Arindam Banerji, Claudio Bartolini, Dorothea Beringer, Abdel Jabbar Boulmakoul, Svend Frolund, Kannan Govindarajan, Alan Karp, Michael Morciniec, Gregory Pogossiants, Chris Preist, Shamik Sharma, David Stephenson, and Scott Williams. Service framework specification, part 1 version 2.0. Technical Report HPL-2001-138, Software Technology Laboratory, HP Laboratories Palo Alto, June 2001.

[5] C. Beeri, A.Y. Levy, and M-C. Rousset. Rewriting Queries Using Views in Description Logics. In L. Yuan, editor, *, New York, USA*, pages 99–108, April 1997.

[6] C. Berge. *Hypergraphs*, volume 45 of *North Holland Mathematical Library*. Elsevier Science Publishers B.V. (North-Holland), 1989.

[7] Fabio Casati and Ming-Chien Shan. Dynamic and adaptive composition of e-services. *Information Systems*, 26(3):143–163, May 2001.

[8] Fabio Casati and Ming-Chien Shan. Models and Languages for Describing and Discovering E-Services. In *Proceedings of SIGMOD 2001, Santa Barbara, USA*, May 2001.

[9] T. Eiter and G. Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM Journal on Computing*, 24(6):1278–1304, 1995.

[10] F. Goasdoué and M-C Rousset V. Lattès. The Use of CARIN Language and Algorithms for Information Integration: The PICSEL System. *IJICIS*, 9(4):383–401, 2000.

[11] Javier Gonzlez-Castillo, David Trastour, and Claudio Bartolini. Description logics for matchmaking of services. In *Proceedings of the KI-2001 Workshop on Applications of Description Logics*, September 2001.

[12] Ian Horrocks and Peter F. Patel-Schneider. The generation of daml+oil. In *Working Notes of the 2001 International Description Logics Workshop (DL-2001)*, August 2001.

[13] Ralf Molitor. Structural subsumption for aln. Technical report, Aachen University of Technology, Research Group for Theoretical Computer Science, March 1998.

[14] Gunnar Teege. Making the difference: A subtraction operation for description logics. In Jon Doyle, Erik Sandewall, and Pietro Torasso, editors, *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR'94)*, pages 540–550, May 1994.